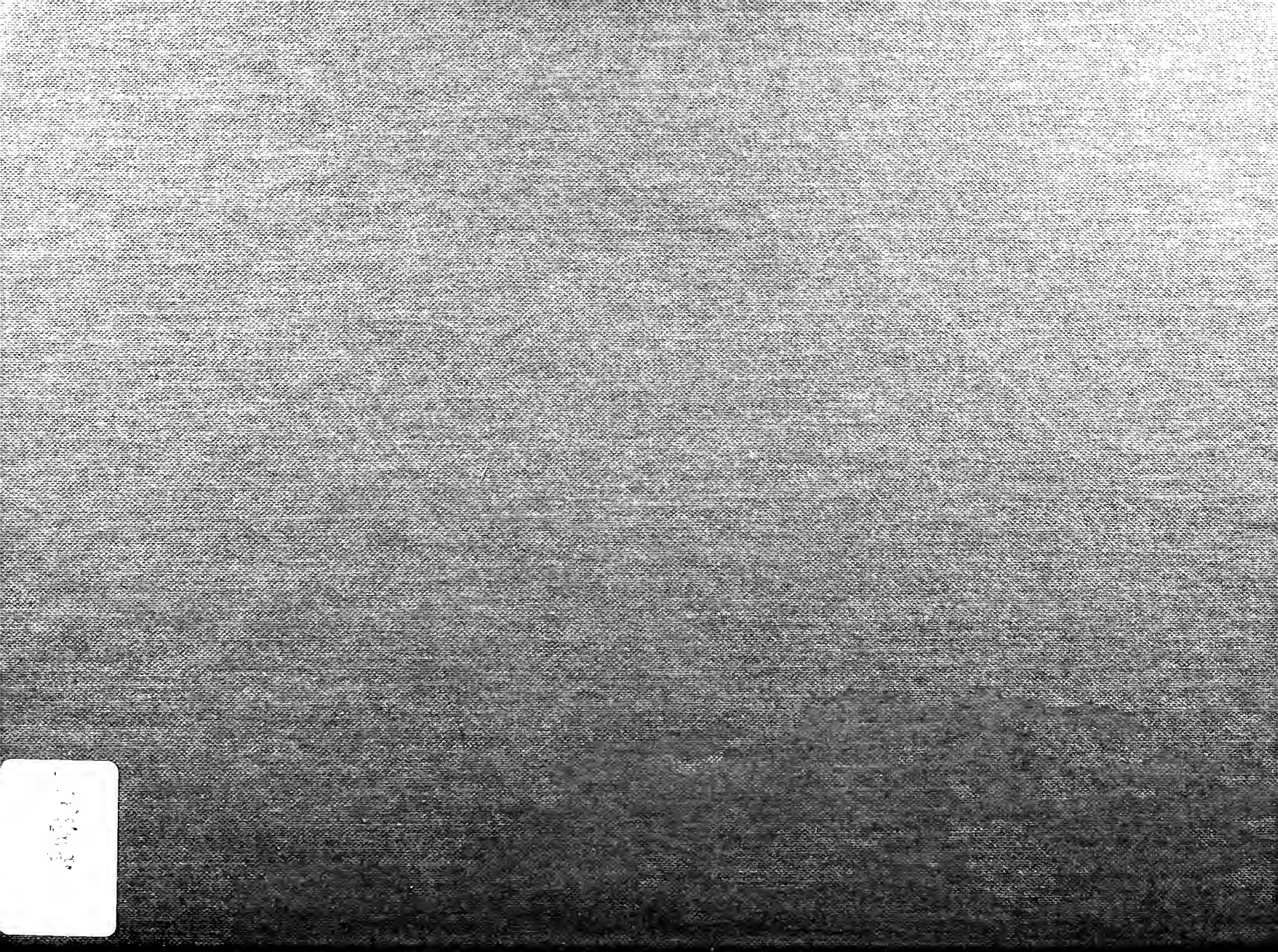


MASSACHUSETTS
LIBRARIES

INFOPLEX: RESEARCH IN A HIGH-PERFORMANCE DATABASE COMPUTER

Stuart E. Madnick
Meichun Hsu

April 1986

CISR WP No. 136
Sloan WP No. 1776-86

Center for Information Systems Research
Massachusetts Institute of Technology
Sloan School of Management
77 Massachusetts Avenue
Cambridge, Massachusetts, 02139

INFOPLEX: RESEARCH IN A HIGH-PERFORMANCE DATABASE COMPUTER

Stuart E. Madnick
Meichun Hsu

April 1986

CISR WP No. 136
Sloan WP No. 1776-86

Published in *IEEE Database Engineering*, Vol. 9, No. 1, March 1986.

Center for Information Systems Research
Sloan School of Management
Massachusetts Institute of Technology

INFOPLEX: RESEARCH IN A HIGH-PERFORMANCE DATABASE COMPUTER

Stuart E. Madnick
Massachusetts Institute of Technology

Meichun Hsu
Harvard University

1. Summary

INFOPLEX is an on-going research effort pursuing technologies for large-scale information management for the 1990's. The goal is to achieve orders of magnitude improvement in speed performance, availability and cost-effectiveness for computers used in information management through fundamental architectural changes. This document provides an introduction to the approaches of the INFOPLEX research project, as well as a summary of the recent research results.

2. Motivation

Due to their enormous storage capacity and processing speed, computers have become a dominating tool for managing information in an information society. The notion of *information utility* was proposed in [HM77, Madnick75b, Madnick77] which described a community in which personal computers and large scale computers are connected to a complex of *information nodes* that provides information services to the community (Figure 1). This view of the future is becoming increasingly plausible, particularly in light of the revolution of the computer industry in the recent past.

To provide information utility, large shared databases are inevitable for a variety of economic and technical reasons [Madnick79]. A computer serving as an information node must satisfy the requirements of high performance, high capacity and high reliability. In [Madnick79] it was envisioned that information nodes in 1990's would be required to have the capability of processing tens or even hundreds of thousands of requests per second (versus around a thousand requests per second on the largest computers today [Scrutchin85]), handling in excess of one hundred billion bytes of on-line structured data, and having the appearance of being operational continuously round the clock.

It has been argued that it would become increasingly costly to improve the speed of the conventional single-processor computer system. One avenue, therefore, to realize the information processing requirements of the 1990's is to seek for changes in computer architecture. Research work in this direction has been categorized as adopting one of the following four approaches [HM77]: (1) new instructions through microprogramming, (2) intelligent controllers, (3) dedicated minicomputers for database operations ("backend computers"), and (4) specialized database computers. In the recent past, commercial database machines, such as Britton Lee's IDM-500 and Teradata, have begun to appear.

3. The INFOPLEX Approach

INFOPLEX is a computer system with special hardware and software architecture dedicated to large-scale information management. The hardware architecture being pursued is a structured multi-processor complex, comprising of up to thousands of micro-processors. The software architecture of the system emphasizes functional decomposition and intelligent synchronization and coordination algorithms to achieve the highest degree of parallelism and robustness.

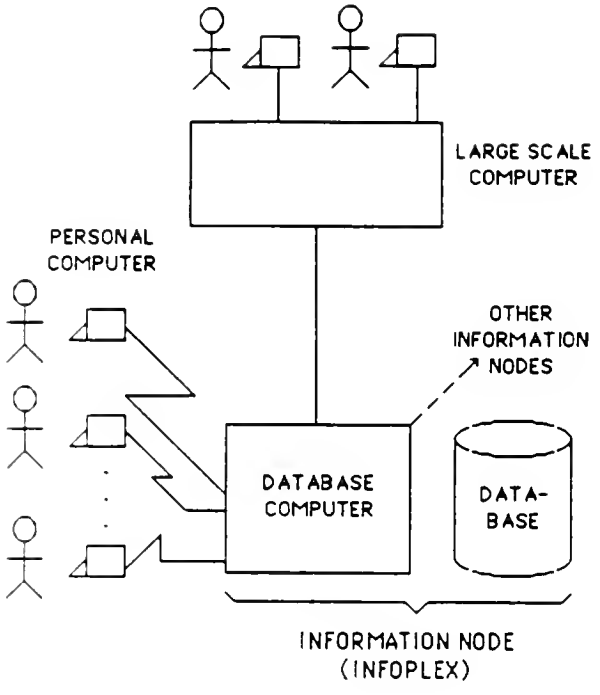

Figure 1.
A Future Trend of Computing:
The Information Utility

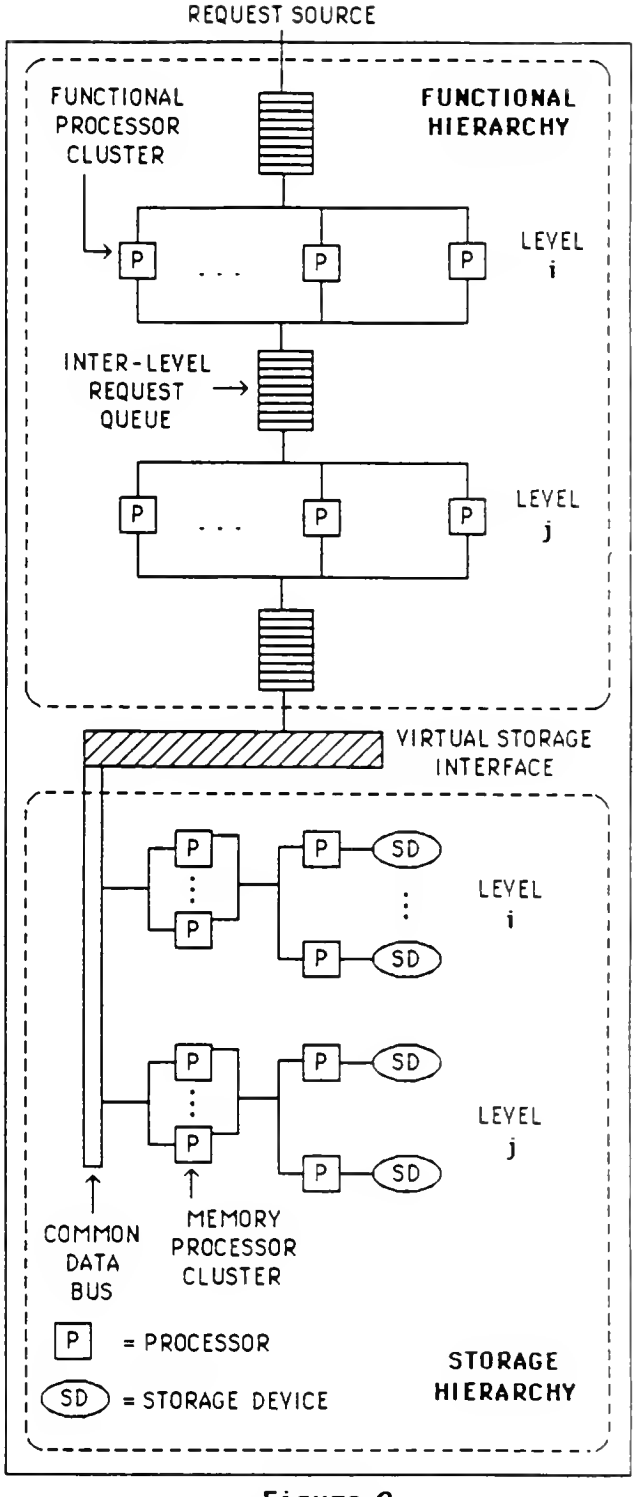

Figure 2.
INFOPLEX Database Computer
Conceptual Organization

3.1. The Architecture of INFOPLEX

The conceptual organization of the INFOPLEX architecture is shown in Figure 2. The architecture design goal of INFOPLEX is to offer large processing capacity economically, reliably and modularly. The trend in hardware technology motivates the use of a large number of microprocessors to take advantage of the cost-effectiveness of the general-purpose microprocessors. In INFOPLEX, these processors are connected via a two-level *bus hierarchy*. The bus architecture is chosen for its simplicity. However, to alleviate the limitation on the number of processors that a single bus can handle, multiple single-bus processor clusters are connected together via a second-level bus, the *global bus*. With such a bus hierarchy and a special high-throughput bus protocol, this architecture is expected to be able to connect a large number of processors together to achieve the required processing capacity.

Central to the coordination of INFOPLEX is the notion of *distributed control*. To attain high performance and high availability, one of the principles followed in the design of INFOPLEX is to have activities of the system coordinated through *distributed control algorithms*. Distributed control algorithms allow components of the system to perform relatively independently of each other by executing predefined protocols without relying on a central coordinator. The use of a central coordinator is often the source of performance and reliability bottleneck problems in a system. Distributed control algorithms are needed to eliminate such a bottleneck. This principle for design is manifested in both of the two major components in INFOPLEX: the *functional hierarchy* and the *storage hierarchy*.

INFOPLEX provides a very large virtual storage that exploits reference locality, is dynamically managed, and does away with centralized memory mangement control. The storage devices are organized into a *storage hierarchy*. The higher levels of the storage hierarchy utilize fast but more expensive storage devices, while the lower levels slower but less expensive devices. The objective is to employ intelligent memory management algorithms for migrating data between levels of the storage hierarchy, achieving the goal that a very large percentage of the data referenced can be found in the higher levels of the storage hierarchy. To perform memory management functions such as memory map, storage allocation and data migration reliably and in a distributed manner, each storage level is also controlled by multiple microprocessors that implement intelligent memory management functions.

The INFOPLEX storage hierarchy is specifically designed to be able to handle any type of storage devices. Thus unlike some other database computer designs which may be specialized to a particular type of storage devices, INFOPLEX can adapt to the changing application needs as well as take advantage of new technological advances in storage devices.

The functional processor clusters are collected into a *functional hierarchy*, each level of the functional hierarchy corresponds to a single-bus processor cluster. The storage hierarchy therefore appears to the functional hierarchy as a very large and intelligent virtual storage. The functional hierarchy therefore dedicates its effort to performing information management tasks, such as message processing, security checking, query decomposition, and internal organization of the data in the database.

The first step towards achieving distributed control in the functional hierarchy is *functional decomposition*. The complex information management function is broken into small tasks. These tasks are arranged in a pipeline manner where each task, representing a stage in the pipeline, may be performed concurrently with other stages. Each stage is assigned to a cluster of processors which comprises a *level* of the functional hierarchy. Within a level, processors perform just one task, with multiple instances of the task being processed in parallel by all processors in that level, thereby gaining further concurrency. Since processors within the same level perform the same set of tasks, intra-level communication/synchronization is more economically achieved through shared memory, while high-throughput inter-level buses are provided for communication among tasks.

In summary, the INFOPLEX architecture is motivated by the need to provide for large processing and storage capacity reliably and economically. The consequence of adopting a structured multiprocessor complex as the basic architecture is the need for these processing and storage components to be organized intelligently and coordinated through distributed algorithms to eliminate potential performance bottleneck.

4. INFOPLEX Recent Research Results

4.1. Research Objectives

Within the above architectural framework, research in INFOPLEX is conducted with the following research objectives:

(1) Discover and define efficient distributed control algorithms and their formal properties;

(2) Study and verify the nature of locality of references in databases for measuring effectiveness of storage hierarchy;

(3) Identify and construct relevant performance evaluation methodologies and apply them to predict INFOPLEX performance and to uncover potential bottlenecks; and

(4) Experiment through software and hardware test vehicles.

4.2. Distributed Algorithms

We describe two particular research efforts aiming at discovering algorithms that are suitable for controlling activities in INFOPLEX: the first effort studies algorithms for controlling migration of data in the storage hierarchy; the second aims at increasing parallelism in database concurrency control and reducing overhead.

In [Madnick75a and LM79], algorithms for data migration in a generalized storage hierarchy are identified. Specifically, data migration algorithms can be classified along two dimensions, both concerning the problem of the maintenance of the LRU (Least Recently Used) stack: the first addresses the treatment of references to the virtual storage from functional processors, the second addresses the treatment of references to levels of the storage hierarchy due to overflow placement.

Along the first dimension, two alternatives are possible: global LRU, which considers every reference to the virtual storage from the functional processors a reference to all levels of the storage hierarchy as far as the maintenance of the LRU stacks at each level is concerned; and local LRU, which only updates the LRU stack at levels of the storage hierarchy that are needed to satisfy the virtual storage reference from the functional processors. Along the second dimension, there are also two alternatives: static overflow placement (SOP) and dynamic overflow placement (DOP). Under SOP, overflow of a page from a higher level i of the storage hierarchy (i.e., closer to the functional hierarchy) to a lower level j of the storage hierarchy is not considered a reference to that page at level j, unless the overflown page is not resident at level j at the time of overflow. Under DOP, however, the overflown page is always considered as being referenced at the lower level, causing an update to the LRU stack at level j.

Two desirable properties significant to the data migration algorithms are also identified: multi-level inclusion (MLI) and multi-level overflow inclusion (MLOI). A storage hierarchy satisfies the MLI property if every page resident at a level i is also resident at the next lower level $i+1$. On the other hand, MLOI is satisfied if every page *overflown* from level i is also found to be resident at level $i+1$ at the time of overflow. These two properties enable the overflow handling algorithms to be simplified considerably and have important implications on data availability. It is proven in [LM79] that only global LRU algorithms can achieve MLI and MLOI. In addition, depending on whether dynamic or static overflow placement algorithm is used, in order for these properties to hold, there is a specific constraint on the size of the lower level of the storage hierarchy in comparison with that of the higher level. The precise nature of the size

requirements is also shown in [LM79].

Using the formal results described above, the design of the INFOPLEX data storage hierarchy is described in [LM79a, LM79b, LM79c, Abraham79, and Madnick80].

In [HM83], a database concurrency control algorithm is identified which aims at reducing synchronization needs among data partitions. In a large database, the data can often be found to be organized in an information hierarchy, with some transactions updating raw data, some transforming the raw data and updating the partition that contains the derived data, and some transforming the first-level derived data and updating the second-level derived data. When the database is partitioned accordingly and distributed to multiple processors, it is desirable that inter-partition synchronization be minimized.

The hierarchical concurrency control algorithm described in [HM83] takes advantage of such a hierarchical structure of the database and leads to protocols which allow the transactions updating one partition to proceed without interfering with transactions updating another partition, thereby minimizing inter-partition synchronization. The protocols described and proven in [HM83] were further extended in [Hsu83] to allow certain degree of cyclic accesses in hierarchically partitioned database.

The above hierarchical algorithms are timestamp based concurrency control algorithms. However, the underlying principles of the algorithm can be abstracted to apply to two-phase locking based algorithms. In [HC85], the adaptation of the hierarchical timestamp algorithm to partitioned two-phase locking is described. This generalization broadens the applicability of the theoretical results.

4.3. Database Locality

The INFOPLEX data storage hierarchy - as well as many file servers, database machines, and DBMS packages - employ dynamic buffering techniques that rely upon pragmatically plausible, but often unproven, database locality for success. The study of database locality provides a theoretical framework for measuring load placed on a memory system by a sequence of requests for access to data. A consistent measure of load at the logical, internal and storage stages of database processing was needed to facilitate the study and comparison of alternative DBMS designs and database structures.

In [Madnick73], notions of temporal and spatial localities are defined. Temporal locality describes the clustering of references along the time dimension: if a data element is referenced at time t, then it is likely to be referenced again shortly after t. Spatial locality, on the other hand, refers to the clustering of references along the space dimension: if a data element d is referenced at time t, then it is likely that some data element d' in the *vicinity* of d is referenced at time $t+1$.

This first attempt to approach locality formally was followed by [McCabe78, Robidoux79] in which reference strings obtained from an existing application system was analyzed for purpose of identifying reference locality. These empirical works further illuminated the need for a theoretical framework for the study of database locality.

In [Moulton86], a theoretical foundation for database locality has been developed using models that are close analog of program locality working set models, but applicable at all stages of database processing - logical, internal, and storage. A two dimensional model that incorporates both temporal and spatial locality effects, with separately adjustable spatial and temporal parameters, is defined. The model reduces to the pure temporal model in the limiting spatial case. This model enables one to measure the locality of a given reference string. Work is currently being performed in applying the above model to measure locality of existing database systems and examining the theoretical properties of the model.

4.4. Performance Evaluation

In this part we describe research efforts directed towards evaluating the speed performance of INFOPLEX. Early performance modeling through simulation for the INFOPLEX storage hierarchy was reported in [LM79d, WM81]. Due to the high level of concurrency being simulated, these early simulation efforts were very costly and it became clear that it was necessary to build analytical evaluation tools that are suitable for a distributed architecture. However, the very nature of the system having *unbalanced asynchronously-spawned (UAP) parallel tasks* violates the flow balance requirement of classical queueing theory.

In [WM84a], an analytical performance evaluation methodology for *flow-unbalanced* networks is described. The method enables a flow-unbalanced queueing network to be transformed into an "equivalent" flow-balanced queueing network through decomposition. For flow-unbalanced *open* queueing networks, this transformation enables the key performance measures to be computed by directly applying the classical queueing theory. For flow-unbalanced *closed* queueing networks, however, the transformation is feasible only when conditions of *network stability* are satisfied. These conditions are formally derived and readily computable given the parameters of a flow-unbalanced closed queueing network. In addition, an efficient iterative procedure for estimating the key performance measures of a flow-unbalanced closed queueing network is developed. The procedure is based on Buzen's convolution algorithm which efficiently computes the normalization constant in the product form solution of a classical closed queueing network.

The solution methodology for flow-unbalanced queueing networks is applied to the INFOPLEX storage hierarchy [WM84b and WM86]. The necessary and sufficient conditions for the INFOPLEX storage hierarchy to be stable when modeled as a closed queueing network are identified. Furthermore, an algorithm has also been devised to test whether a design alternative of the INFOPLEX storage hierarchy will be stable.

4.5. Experimental Test Vehicles

In this section, the approach taken to build a test vehicle for INFOPLEX is described. It consists of a multi-microprocessor hardware test vehicle and a simulated software test vehicle.

4.5.1. Software Test Vehicle

The purpose of the software test vehicle (STV) project is to test out the preliminary designs of the functional hierarchy [Hsu80] and the storage hierarchy [LM79a] in software emulation on contemporary hardware before committing them to hardware prototypes. The first approach was to emulate the parallelism of the target architecture on a conventional single-processor computer. The STV project was carried out on an IBM 370 mainframe and was implemented in PL/1. The project consisted of three parts: a functional hierarchy STV [BM81, Hsu82a, Lee82, Lu82], a storage hierarchy STV, and a hardware emulator, called Shell [To82].

The preliminary design implemented in STV adopts a software paradigm composed of modules. The modules in STV are distributed to functional levels in the emulated functional hierarchy based on the nature of the task performed by the modules. The inter-level module invocation is performed through message passing, emulated by "Shell", and no argument passing through shared memory variables is allowed. From the STV experience, it appears that there is a need to identify a software paradigm, or a functional decomposition methodology, that is more sensitive to the distributed nature of the target hardware, and subject the design of the functional modules to the paradigm. Work is currently being performed to identify such a methodology.

The cost of software emulation of a concurrent system architecture employing high degrees of parallelism and pipeline processing on a conventional mainframe renders it impractical to conduct extensive experiments. A more effective method for experimenting with the concurrent functional software is to adopt parallel computers, such as off-the-shelf micro-computers inter-

connected through a high bandwidth network. Work is currently being pursued to set up such a more advanced test bed environment for further software experiments.

4.5.2. Hardware Test Vehicle

For the hardware test vehicle, the purpose is to demonstrate the feasibility of the INFOPLEX multi-level multiprocessor hardware architecture. The first task is to build a one-level shared-memory multiprocessor system. Several issues are investigated. The first issue is the choice of the processor. To this end, off-the-shelf microprocessors are evaluated [TG81, GT83a, GT83b, GT83c] and trends in evolution monitored closely [GT85b]. Methodologies for evaluating microprocessors are reported in [GAT81, GT82]. In particular, in [GT82], a hierarchical approach to evaluating microprocessors is also proposed.

The issue of multiprocessor interconnection scheme is studied in [GT80, GST80, Gupta82] In particular, the *pended bus architecture* utilizing the split transaction protocol is shown to be especially effective in minimizing potential contention on the bus and therefore able to support a large number of processors on the bus. The split transaction bus protocol allows a processor to relinquish control of the bus when its memory request is being serviced at a memory module, enabling another processor or memory module to obtain control of the bus in the mean time to transfer another request or data over the bus. The protocol therefore allows the effective bus throughput to be increased. Performance of the pended bus architecture is studied in detail to resolve relevant design decisions at the implementation level. These results are reported in [Gupta82, GT82, GT84, GT85a], and are used in designing the bus interface unit (BIU) that connects processors and memory modules to the intra-level bus of INFOPLEX.

Another shared memory multiprocessor issue is the cache consistency problem. In [AM81], this problem is studied in detail and simulation conducted to evaluate the performance of various schemes. In the prototype multiprocessor system, the cache consistency problem is avoided by not allowing data segments to be cached.

The prototype one-level shared-memory multiprocessor system is completed, together with its kernel software, i.e., the *local operating system* [TAL86]. Work is being performed in replicating the one-level prototype and connecting them through a global bus (i.e., the inter-level bus). The same BIU design will be utilized in connecting a level to the global bus. The local operating system will also be extended to handle inter-level communication protocols.

5. Conclusion

The purpose of the INFOPLEX research project is to advance the technologies for information management through both basic and experimental research within the framework of a structured multi-processor architecture. In this document, a summary of the recent research results of the project is reported. The research falls in the following categories: (1) distributed algorithms, (2) database locality, (3) performance evaluation methodology, and (4) experimental test vehicles. These works lay the foundation for future research and implementation efforts, while additional work is needed to refine and apply the theoretical results and resolve detailed design and implementation decisions.

6. Technical Reports and References

[Abraham79]: Abraham, M. 'Properties of reference algorithms for multi-level storage hierarchies,' Master's Thesis, M.I.T. Sloan School of Management, 1979.

[AM81]: Abdel-Hamid T.K. and Madnick, S.E. 'A study of the multicache-consistency problem in multi-processor computer systems,' *Proceedings of the Sixth Workshop on Computer Architecture for Non-Numeric Processing*, June 1981.

[BM81]: Blumberg, B. and Madnick, S.E., 'INFOSAM: A sample database management system,' NTIS No. AD-A116-593, December 1981.

[GAT81]: Gupta, A., Abdel-Hamid, T. and H.D. Toong, 'A Comparison of analytic and simulation models,' M.I.T. Sloan School of Management, December 1980.

[GST80]: Gupta, A., Strommen, S.O. and H.D. Toong, 'Evaluation of multimicroprocessor bus architecture,' M.I.T. Sloan School of Management, May 1980.

[GT80]: Gupta, A., and H.D. Toong, 'Interactive multimicroprocssor performance systems,' Sloan School of Management, April 1980.

[GT82]: Gupta, A., and H.D. Toong, 'Enhanced concurrency in m-n multiprocessor systems,' *Proceedings of the IEEE Third International Conference on Distributed Computing Systems*, October 1982.

[GT83a]: Gupta, A., and H.D. Toong, (eds.) *Advanced Microprocessors*, IEEE Press Selected Reprint Series, IEEE Computer Society, IEEE Press, New York, NY, 1983.

[GT83b]: Gupta, A., and H.D. Toong, 'An Architectural comparison of 32-bit microprocessors,' *IEEE Micro*, Vol. 3, No. 1, February 1983. (republished in *Microprocessors and Microcomputers*, IEEE Press: New York, NY, 1984)

[GT83c]: Gupta, A., and H.D. Toong, 'Microprocessors - the first twelve years,' *Proceedings of the IEEE*, Vol. 71, No. 11, November 1983.

[GT84]: Gupta, A., and H.D. Toong, 'Microcomputers in industrial control applications,' *IEEE Transactions on Industrial Electronics*, Vol. IE-31, No. 2, May 1984.

[GT85a]: Gupta, A., and H.D. Toong, 'Increasing throughput of multiprocessor configurations,' *IEEE Transactions on Industrial Electronics*, August 1985.

[GT85b]: Gupta, A., and H.D. Toong, 'Trends in Microcomputers,' in *International Handbook of Information Technology and Office Systems*, A.E. Cawkell (ed.) North-Holland Publishing Company: Amsterdam, Netherlands, 1985.

[Gupta82]: Gupta, A., 'Performance modeling of multimicroprocessor systems,' *Proceedings of the South-East Asia Regional Computer Conference*, Kuala Lampur, Malaysia, September 1982.

[HC85]: Hsu, M. and Chan, A. 'Partitioned two-phase locking,' *Proceedings of the First International Workshop on High Performance Transaction Systems*, September 1985.

[HM77]: Hsiao, D.K. and Madnick, S.E., 'Database machine architecture in the context of information technology evolution,' *Proceedings of the Third International Conference on VLDB*, October 1977.

[HM83]: Hsu, M. and Madnick, S.E. 'Hierarchical database decomposition: A technique for database concurrency control,' *Proceedings of 2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, March 1983.

[Hsu80]: Hsu, M. 'A preliminary architectural design for the functional hierarchy of the INFOPLEX database computer,' NTIS No. AD-A102-924, November 1980.

[Hsu82]: Hsu, M. 'FSTV: The software test vehicle for the functional hierarchy of the INFOPLEX database computer,' NTIS No. AD-A116-591, January 1982.

[Hsu83]: Hsu, M. 'The hierarchical decomposition approach to database concurrency control,'TR M010-8312-16, M.I.T. Sloan School of Managemenet, December 1983.

[Krakauer80]: Krakauer, L. 'Virtual information in the INFOPLEX database computer,' Master's Thesis, M.I.T. Sloan School of Management, 1980.

[LM79]: Lam, C.Y. and Madnick, S.E., 'Properties of storage hierarchy systems with multiple page sizes and redundant data,' *ACM Transactions on Database Systems*, Vol 4, No. 3, September 1979.

[LM79a]: Lam, C.Y. and Madnick, S.E., 'Intelligent memory system architectures - research directions,' NTIS No. AD-A073-485, June 1979

[LM79b]: Lam, C.Y. and Madnick, S.E., 'The IMS data storage hierarchy - DSH-I,' NTIS No. AD-A073-375, June 1979.

[LM79c]: Lam, C.Y. and Madnick, S.E., 'The IMS data storage hierarchy - DSH-II,' NTIS No. AD-A073-376, June 1979.

[LM79d]: Lam, C.Y. and Madnick, S.E., 'Simulation studies of the DSH-II data storage hierarchy system,' NTIS No. AD-A074-503, June 1979.

[Lee82]: Lee, J. 'Virtual information facility of the INFOPLEX software test vehicle (Part I),' NTIS No. AD-A116-503, May 1982.

[Liu82]: Liu, D. 'N-ary level design of the INFOPLEX software test vehicle,' Bachelor's Thesis, M.I.T., 1982.

[Lu82]: Lu, P. 'Virtual information facility of the INFOPLEX software test vehicle (Part II),' NTIS No. AD-A116-502, May 1982.

[Madnick73]: Madnick, S.E. 'Storage hierarchy systems,' TR-105, Project MAC, M.I.T., 1973

[Madnick75a]: Madnick, S.E., 'INFOPLEX - Hierarchical decomposition of a large information management system using a microprocessor complex,' *Proceedings AFIPS 1975 International Computer Conference*, Vol. 44, May 1975.

[Madnick75b]: Madnick, S.E., 'Design of a general hierarchical storage system,' *Proceedings of the International Convention and Exposition of the Institute of Electrical Engineers*, April 1975.

[Madnick77]: Madnick, S.E., 'Trends in computers and computing: the information utility,' *Science*, Vol. 185, March 1977.

[Madnick79]: Madnick, S.E., 'The INFOPLEX database computer: concepts and directions,' *Proceedings IEEE Computer Conference*, February 1979.

[Madnick80]: Madnick, S.E., 'Recent research results on the INFOPLEX Intelligent Memory System Project,' *Proceedings of the International congress on Applied Systems Research and Cybernetics*, December 1980.

[McCabe78]: McCabe, E., 'Locality in logical database systems: a framework for analysis,' Master's Thesis, M.I.T. Sloan School of Management, 1978.

[Moulton86]: Moulton, A., 'The foundation of database locality,' in preparation.

[Robidoux79]: Robidoux, S., 'A closer look at database access patterns,' Master's Thesis, M.I.T. Sloan School of Management, 1979.

[Scrutchin85**]: Scrutchin, T. 'TPF: Performance, capacity and availability,' *Proceedings of the First International Workshop on High Performance Transaction Systems*, September 1985.

[Stortz83]: Stortz, H., Jr., 'A classification scheme for database machine architectures,' Bachelor's Thesis, M.I.T., 1983

[TAL86]: Toong, H.D., Abraham, M. and Linsky, M. 'The design and implementation of a fault-tolerant multiprocessor for the INFOPLEX database computer,' in preparation.

[TG81]: Toong, H.D. and Gupta, A. 'An architectural comparison of contemporary 16-bit microprocessors,' *IEEE Micro*, Vol.2, No. 2, May 1981. (reprinted as a chapter in *Microcomputer Networks*, IEEE Press: New York, NY, 1981)

[TG82]: Toong, H.D. and Gupta, A. 'Evaluation kernels for microprocessor analysis,' *Performance Evaluation*, North-Holland Publishing Company: Amsterdam, Vol., 2, No. 1, 1982.

[TG84]: Toong, H.D. and Gupta, A. 'Hardware feasibility of INFOPLEX design strategy,' Sloan School of Management, May 1984.

[To82]: To, T., 'SHELL: a simulator for the software test vehicle for the INFOPLEX database computer,' NTIS No. AD-A116-592, August 1982.

[WM81]: Wang, Y.R. and Madnick, S.E., 'Performance evaluation of the INFOPLEX database computer using operational analysis,' TR M010-8109-13, M.I.T. Sloan School of Management, September 1981.

[WM84a]: Wang, Y.R. and Madnick, S.E., 'Queueing network systems with unbalanced flows and their applications to highly parallel distributed information systems,' TR M010-8408-14, M.I.T. Sloan School of Management, August 1984.

[WM84b]: Wang, Y.R. and Madnick, S.E., 'Performance evaluation of distributed systems with unbalanced flows: an analysis of the INFOPLEX data storage hierarchy,' TR M010-8109-15, M.I.T. Sloan School of Management, July 1984.

[WM86]: Wang, Y.R. and Madnick, S.E., 'Modeling multiprocessor computer systems with unbalanced flows,' to be published in *Proceedings of the Joint Conference on Computer Performance Modelling, Measurement, and Evaluation*, 1986.

*Note:
Documents distributed through the National Technical Information System (NTIS) can be obtained by writing to: NTIS, 5282 Port Royal Rd., Springfield, Virginia 22161

Acknowledgement: The authors would like to thank the following current and past participants in the INFOPLEX Project for their valuable contribution to the research progress and results reported in this paper:

Current Investigators:

Hoo-Min D. Toong, M.I.T. Amar Gupta, M.I.T.
Richard Ying Yu Wang, University of Arizona

Current Participants:

Mike Abraham Allen Moulton
B.E. Prasad (on a Grant from Indian Government)

Previous Investigators and Participants:

Peter P.S. Chen	David K. Hsiao
Tarek K. Abdel-Hamid	Zeev Bulka
Bruce Blumberg	Benjamin Chou
Larry Krakauer	Earl Goodrich
Chat Lam	Jameson Lee
Peter Lu	David Lui
Edward McCabe	John-Francis Mergen
Susanne Robidoux	Charles J. Smith
Herb Stortz	Svein Ove Strommen
Tak To	Anthony Wang

The INFOPLEX project has been mainly supported by the Space and Naval Warfare Systems Command (formerly the Naval Electronic Systems Command) under contract numbers N00039-78-G-0160, N00039-80-K-0573, N00039-80-K-0498, N00039-81-C-0663, N00039-83-C-0463, N00039-85-C-0571. Some earlier work was supported by a grant from National Science Foundation under contract number MCS 77-20829. The work on the evaluation and implementation of the multiprocessors has been partly sponsored by a grant from I.B.M. The views expressed in this document are those of the authors and should not be interpreted as representative of the official policies, either expressed or implied, of any sponsoring agencies.